---------------------------------------
CIRCULO USUARIOS QL NUMERO 1 OCTUBRE 1988
---------------------------------------

INDICE
======



EDITORIAL

Mi nombre es Salvador Merino, y viendo que en QLave Zaragoza ya no son capaces ni de confeccionar una cosa tan simple como un boletín, he decidido ayudar un poco.

Hay que dejar algo bastante claro, no me hago responsable de ese hatajo de irresponsables con la cara más dura que el cemento que forman la Junta Directiva en Zaragoza.

Las colaboraciones deben ser enviadas a:

Salvador Merino
Ctra. Cádiz, Cerámicas Mary
29640 Torreblanca del Sol
Fuengirola (Málaga)
Teléfono 475043

Prometo, aunque sé que me va a costar dinero de mi bolsillo, devolver los discos un dia o dos después de su recepción (por favor, cuando reciban los discos de retorno, no borren sus ficheros hasta que vean publicados sus articulos).

¿ Cuál es el formato de las colaboraciones ?

Son ficheros QUILL con el margen izquierdo a 0, el sangrado a 0, y el derecho a 80. Máximo 45 lineas por página. No pie de página. Solamente usar caracteres normales (No negrita, subrayado,.....).

¿ Qué se puede enviar ?.

Muy simple cualquier cosa (Cartas, preguntas, programas, trucos, comentarios, criticas, anuncios,.... se imprime de todo, e incluso se aceptan cartas a la novia y felicitaciones).

Si QLave desapareciese algún dia (cosa muy probable), las colaboraciones no se habrian hecho en vano, pues pasarian a engordar el boletín de mi Club privado, C.U.Q. (CIRCULO de USUARIOS QL). Formar parte de mi Club, es algo verdaderamente dificil, solamente se admiten un número muy limitado de colaboradores.

Adjunto a este fichero QUILL he enviado tres programas cedidos para la libreria

(si alguien consigue echarle el guante a alguno, puede sentirse afortunado), y son :

- Superforth Calculator v1.1.- Como su nombre indica es una calculadora de anotación polaca inversa que usa todo el paquete aritmético de coma flotante del QL. Se envia también listado fuente por si alguien está interesado en agregar algo o perfeccionarlo. Autor: Salvador Merino

- Ratón.- Un toolkit que agrega varios comandos al Superbasic para el manejo del ratón de Inves en nuestros programas (próximamente versión en Forth). Autor : José Carlos de Prada.

- Retrato Robot.- Un juego en Superbasic compilado con el Qliberator que consiste en hacer el retrato robot del asesino que vimos al empezar el juego. Autor: José Carlos de Prada.

S. Merino

NOTICIAS DE FINALES DE PRIMAVERA

- El prototipo de QUANTA es un nuevo modelo de QL basado en el original, pero con una nueva PCB que nada tiene que ver con la original y sin sus defectos (el THOR I usaba la misma PCB que el QL original). Solamente existen tres PCB actualmente. Tiene su MC 68008, todo lo que tiene un QL (excepto la opción a MDVs), interface paralelo y disco.

- El CST THOR XVI, el otro basado en el MC 68000, no es totalmente compatible con el QL debido a que el Keyrow no es compatible con el QL (Y el emulador de Keyrow solamente funciona con los programas en Superbasic/compilados), pero puede correr casi todo el software del QL (p.e.: la Qram, Taskmaster, swooper, keydefine,... son programas que no corren en un Thor). El sistema operativo se llama ARGOS.
El THOR XVI posee ya de un emulador de Macintosh pudiendo lanzar los programas Macintosh en multitarea (el problema es ¿ Qué Modelo de Macintosh emula ?). Aunque existe el problema de que la definición de la pantalla en el Mac es mucho más grande que en los QLs, el Thor XVI tiene un nuevo modo, MODE 12, que permite 16 colores en una pantalla de 256x256 pixel. Esta pantalla es producida usando las posibilidades extra del Hitachi HD6445 video controller. Pero ese chip está capacitado para una resolución de 640x400 pixel en 16 colores.
El problema que tiene el THOR actualmente es su precio y la producción de solamente 500 unidades mes. Aún asi, creen que la URSS y los paises del Este de Europa podrian estar muy interesados en el THOR. Por ahora el THOR XVI se está vendiendo bien en el norte de Europa.
Se ha anunciado la aparición de un nuevo THOR basado totalmente en el MC 68020.
- El Atari 1040 ST o Mega ST se puede convertir en QL compatible con el QL Emulator on the Atari ST. Consiste en una tarjeta hardware que contiene piezas especificas QL, y un disco con el sotfware necesario y una copia exacta de la ROM de un QL (Es la única manera de burlar los derechos de Amstrad, cualquier usuario puede hacer una copia de su ROM en disco). Este emulador permite usar los periféricos standard de un Atari ST (MIDI, Paralelo, disco, memoria extra....)
- El nuevo prototipo de Tony Tebby es ampliar a los PCs en FUTURA. La Card-Futura consiste en una tarjeta que contiene un MC 68010 y los sistemas operativos OS/9 y SMS 2 (el sistema operativo compatible QDOS original del FUTURA). La tarjeta tiene espacio suficiente para 4 Mbutes de RAM, opción MC 68020 o un Transputers, expansión interna para periféricos ATARI (SCSI hard disk),....
En Quanta creemos que es un error, pero un PC + Card Futura es más barato que un THOR XVI. Es muy posible que Sandy planee usar los beneficios de esta tarjeta para el lanzamiento del Super Futura (basado en el MC 68020).
- La limitación del paquete aritmético de coma flotante del QL ya no es tanta con el nuevo paquete QPAC 1 que agrega al sistema Qram una calculadora con 40

digitos de precisión.
- Sandy posee un nuevo prototipo de la Megaboard, la SuperDuperQboard.
- ABC anuncia la Mega-Ram que consiste en una ampliación de la Ram por cambio de CPU hasta 3 MB.
- Se están anunciando Clubs de QL/THOR. P.e.: Quanta, S.U.B, Thornado-Systems Club, etc...
- Continuan saliendo titulos nuevos de Software en cantidades dificiles de absorver.

NOTICIAS DE SEPTIEMBRE

He guardado las malas noticias para el final, y creo que son para llorar.

La firma en que todos teniamos puesto el futuro del QL, CST, ha quebrado. Lo que significa que el futuro del THOR XVI está en el aire, y solamente lo puede salvar la firma Dansoft y PDQL haciendose cargo de la continuación de su producción. Si el THOR XVI no sigue produciendose, los paises interesados del Este dejarian de estar interesados y el THOR XVI nunca seria fabricado en el Este de Europa para venderse en el Oeste de Europa.

La otra cara de la moneda, Tony Tebby, posiblemente va a parar la producción de productos QL. Seria una lástima, pues los últimos productos sin terminar son muy buenos (aún no ha sido confirmado).

Y aún hay más, el prototipo de Quanta y el emulador QL para Atari ST necesitan piezas especificas QL, y esas piezas se están agotando debido a que Amstrad no ha respetado los 8 años minimos de piezas de recambio que habia firmado, pues en los últimos dos años no ha fabricado ni una sola pieza (Quanta dice que posee suficientes firmas de usuarios para obligar a Amstrad a fabricar esas piezas).

Todo esto ha pasado antes de la celebración del dia internacional de los usuarios del QL, dia 23 octubre en Bélgica. Habrá que esperar a ver que pasa, pues hemos salido de situaciones peores en el pasado.

Pase lo que pase, voy a continuar con mi QL por lo menos dos años más o el tiempo que haga falta, pues ningún ordenador de todos los actuales ofrece garantias para el futuro, y con mi QL puedo hacer de todo.

Aún no he comprado el Sinclair QL World de octubre y no he recibido el Quanta de septiembre, pero creo que el QL va a seguir felizmente muerto como en los últimos 3 años (quiero decir que si hemos tenido una agonia de tres años, podemos continuar asi otros tres años).

Salvador Merino
Fuengirola (QLave-154)

ERASE UNA VEZ

La mayoria de todos nosotros conocemos la historia de nuestro cacharro negro con teclas, pero creo que es hora de recordar algunas cosillas del pasado.
Cuando me meti en el ajo, era solamente un usuario de juegos en 1979, y un ordenador de oficina marca Atari con 16K de RAM sin unidad de disco costaba un riñón, y los juegos venian en cartucho ROM (ahora te venden el mismo que costaba 200.000 # por 4.000 # y sin usar). Pero apareció Sinclair y los precios empezaron a bajar. Compré un ZX Spectrum 48k en diciembre 83, y conocí el Basic por la mano de la revista ZX. Unos meses más tarde ví el lanzamiento del QL, y las características del Cacharro me gustaron muchisimo. En 1984 compré la ZX printer, el interface I, II, interface indescomp y una unidad de MDV para el Spectrum, y decidí ahorrar para comprar un QL para la navidad de 1985. Aún no me habia enterado de como se usa el QL cuando de golpe y porrazo dicen las revistas que Amstrad ha comprado Sinclair y va a dejar de fabricar el QL. Después dicen que un grupo de empresas están intentando comprar los derechos de producción del

QL, pero no tienen éxito.
La última noticia era que dos empresas, CST y SANDY, estaban trabajando en dos SuperQLs basados en el 68.000. LLega el verano y desaparecen todas las publicaciones del QL como si nunca hubiesen existido (menos Sinclair QL World). Pude comprar algunos libros y conocer el assembler 68.000 que me pareció más fácil de usar que el Assembler Z80. Compré el Assembler Worbench por mediación de QLhard y me localizaron los de QLave.
Cuando conocí QLave, estaban sufriendo la euforia de la llegada del FUTURA, pero ese bicho nunca llegó y el Club empezó a desmoralizarse y desmoronarse (el verano fue horroroso debido a la ofensiva de los PCs para apoderarse del mayor número posible de usuarios).
Cuando todo parece que está perdido, aparece en escena el segundo Super QL, el CST THOR XVI. Ahora todo va a cambiar (o eso creo) y el mundo del QL tiene garantizado un sucesor, pero el Apple Macintosh, el Atari ST, el Commodore Amiga y otros cientos de máquinas basadas en un 68.000 u otros microprocesadores ultra modernos, son rivales de campeonato.

¿ Podrá el THOR XVI competir con los otros 68.000 ?

La respuesta es si y no. Como único 68.000 europeo tendrá mucha aceptación en nuestro continente. En precio está algo carillo, pero es un modelo pensado para competir en el mercado profesional de alta calidad y prestaciones, que necesita un buen sistema operativo multitarea económico. Un nuevo 68.000 sin software ni hardware estaría condenado al fracaso desde el primer dia, pero en este caso particular tiene una gran cantidad de software y hardware que pertenece al Legendario y aún poderoso QL.
Creo que debería haberse sacado un modelo económico para el hogar igual que los ST y Amiga pequeños. Pero parece que esa opción se la han dejado a Amstrad. Después de todo Amstrad nunca saca nada nuevo, pues en sus CPC y PCW estaba basado su éxito en el obsoleto CP/M, y en sus nuevos PCs el éxito está basado en el anticuado MS-DOS. Aún asi, es muy posible que detrás de este sucesor veamos una gran cantidad de sucesores totalmente compatibles con los anteriores cada vez más potentes y más rápidos (y más caros).

Aunque a primera vista no es un QL-compatible, el Atari 1040 ST podria convertirse fácilmente en un QL-compatible añadiendole la tarjeta QL Emulation on The Atari ST, que vende Strong Computer Systems por 199 libras. El usuario de un Atari ST podria ganar con esa tarjeta un sistema operativo multitarea con una cantidad de software comercial similar a la suya, pero con la diferencia de ser la mayoria de gestión (no juegos).

La mayoria de los socios con QL ampliado usamos el Turbo Xchange v3.87, que es la última versión para el CST THOR. Aunque esta versión no corre en un QL, ha sido modificada por alguien (nunca se cuelga, demostrado en más de 6 horas de trabajo seguidas) y corre muchisimo más rápido que las versiones para los QLs. Tiene comandos y opciones completamente nuevos, corre con parte de la Qram, usa el Toolkit II, es multitarea con el Basic y otros programas, etc.......

Si nuestra máquina con una producción cero ha conseguido seguir viva entre los usuarios y tener todo lo que ahora posee, ¿ Qué pasará ahora que tiene un sucesor en producción ?

¿ Qué pasará con QLave en 1989 ?. La respuesta está aún en el aire, pero si seguimos perdiendo socios al mismo ritmo que actualmente, y los socios no envian colaboraciones y algún que otro programa pequeño escrito en el lenguaje que sea, es casi seguro que en 1989 QLave esté formado, si existe, por los cuatros gatos minimos necesarios que estén de acuerdo por escrito en continuar con el Club (ver los estatutos del Club de 1986, Capitulo V Disolución y liquidación ).

Tal como yo veo las cosas, tiene más futuro el CST THOR y sucesores que nuestro propio Club QLave. Y eso que nuestro Club ha durado más tiempo que algunas revistas famosas. P.e.: ZX nació en Diciembre 83 y murió en Mayo 87 (casi un año después de su desaparición de los kioscos en Málaga, y una agonia lenta por falta de interés en sus últimos articulos).

Pero recordad, el Club aún puede salvarse y la solución está en vuestras manos, pero si creeis que la solución es simplemente cambiar a otro ordenador, creo que no sabeis lo que realmente quereis (con un QL se puede hacer casi de todo).

Ante todo, tengo que aplaudir a todos esos socios nuevos que han enviado sus

colaboraciones publicadas en el número Mayo/junio. Si sigue el ejemplo, el Club se salvará, pero si no, el Club morirá sin remedio.

Creo que esta vez me he pasado escribiendo, pero también deseo poner mi pequeño anuncio :

Chatarrero informático compra ordenadores Sinclair QL Standard, y ofrece 2.000 ptas por uno averiado y 5.000 ptas por uno en perfecto estado (los gastos de envio corren por mi cuenta). Solamente me interesan por ahora 2 o 3 máquinas. Para recibir confirmación los interesados deben escribir a Salvador Merino, Ctra Cádiz Cerámicas Mary, 29640 Torreblanca del Sol, Fuengirola (Málaga), Teléfono 475043 o 474887 los fines de semana.

Salvador Merino
Fuengirola (QLave-154).

CARTAS DE LOS SOCIOS

Antes que nada deseo saludar a todos los socios, pues acabo de incorporarme a QLave. Pienso que como recien llegado mis opiniones sobre el club os pueden resultar de intesrés, por aquello de la objetividad.
Mi primera impresión, debo reconocerlo, no fue excesivamente positiva: pese a haber pagado por giro mi cuota correspondiente al segundo semestre de 1.987 a mediados de Octubre pasado, no he recibido los boletines correspondientes hasta hace una semana.
La segunda no fue mucho más alentadora, pues fui a leer directamente el editorial del mes de Diciembre: el presidente dimite ante la falta de colaboración de los socios y todo ello en medio de una situación económica cuando menos difícil. Os aseguro que mi primera idea fue "¿En dónde me he ido yo a meter?".
Pasado el susto me volví a dedicar a ojear los distintos números de QLave que había recibido, deteniéndome de vez en cuando a leer más atentamente algún artículo. El resultado fue que no solamente se borraron del todo las malas impresiones anteriores, sino que además me fui convenciendo de que solicitar el ingreso en el club había sido todo un acierto y sólo siento no haberlo conocido antes. En mi opinión la gran mayoría de los artículos, comentarios, rutinas, ect. son de primera calidad.
A parte de opiniones estoy dispuesto a aportar todo lo que esté en mi mano para colaborar a la buena marcha del club. Como veis, lo primero que he hecho ha sido sentarme a teclear unas líneas para el boletín. Además me gustaría ofreceros algunas sugerencias para luchar contra los problemas que actualmente padece QLave, según lo que he podido leer. Como soy un recien llegado es muy posible que algunas de mis ideas se hayan planteado ya antes y se hayan desechado por razones importantes; en ese caso basta con que las paseis por alto, si en cambio no es así os agradeceré que las tengais en cuenta, sobre todo los miembros de la junta.
Los problemas principales de QLave, tal como me ha parecido entender, son dos: 1o la situación económica y 2o la falta de nuevos socios.
Con respecto al primer problema he podido comprobar como se ha tomado la iniciativa de aumentar la cuota, solución desagradable, pero seguramente imprescindible. Sin embargo no creo que sea el único punto por el que se deba atacar el problema. Me ha llamado la atención leer unos datos sobre el número de afiliados en los que se podían apreciar unas fluctuaciones al final de cada semestre. La explicación de este fenómeno se encuentra sin duda en el sistema de recaudación de cuotas: al estar basado en la iniciativa de cada socio, se producen retrasos y es posible que incluso descuelgues. Pienso que sería mucho más efectiva, a la vez que cómoda para el socio, la domiciliación de las cuotas en cuenta bancaria, para ello habría que enviar a cada afiliado un impreso de autorización a su banco para el pago de la cuota en las fechas oportunas; una vez rellenos estos impresos y devueltos por el socio, basta con enviarlos a cada banco y este se encarga de los pagos en su momento. Este papeleo, que tampoco es excesivo, hay que llevarlo a cabo sólo una vez; a cmbio pienso que se puede ganar mucho en regularidad de los ingresos y en evitación de pérdidas innecesarias de socios.
Con respecto al segundo problema, creo que la culpa fundamental la tiene el desconocimiento de muchos usuarios de la existencia de QLave. En mi caso me enteré a través de un número atrasado de Todospectrum que casualmente cayó en mis manos. Se que ya se están llevando a la práctica algunas campañas de

captación de socios. Quizá lo más díficil sea llegar hasta los usuarios: es posible que a través de Investrónica se pueda obtener una lista de clientes. En todo caso esa lista debe existir en algún sitio pues Valente envía propaganda por correo a los usuarios del QL.
Otra forma de llegar a los hipotéticos futuros socios podría ser el envío a Sinclair QL World de una nota informativa de nuestra existencia y de nuestras actividades (los Suecos lo han hecho en el número de Febrero en la sección QL SCENE y hace unos meses lo hicieron los Belgas en la sección OPEN CHANNEL). Insisto en este punto ya que he podido comprobar que esta revista se vende mucho, al menos aquí en Madrid, y deberíamos aprovechar su difusión.
Como no quisiera limitarme a las cuestiones burocráticas y además me considero, aunque principiante, un vicioso de la programación y un enamorado del FORTH, os envío una colaboración aparte sobre algunas extensiones en este lenguaje.

José Carlos de Prada
MADRID (QLave 216)

RESPUESTAS A LAS PREGUNTAS DE LOS SOCIOS

RESPUESTA a la PREGUNTA de Marco Cruz en QLave Mayo/Junio 88

El Abacus v2.21 Español no corre en QL ampliados, pero puede correr con tu chapuza o con la Qram con 32 K de espacio para datos. Te recomendaria que obtuvieses la v2.3, v2.35, o simplemente la Turbo Xchange v3.87 pirata (Adaptación ilegal al QL).

Hay muchos programas que no corren en QLs ampliados. Por ejemplo : el D-DAY corre en modo 896 K sin gráficos y en modo 128 K se cuelga durante la carga, pero si usamos el programa Reset 128 K de la Qflash (un pequeño programa en Superbasic de 6 lineas que carga desde DATAs un pequeñisimo programa en código máquina), el QL ampliado con la Trump Crad (o lo que sea), el QL se resetea, y se convierte en un QL 128 K tal y como nos lo vendió Inves. Por lo tanto, no hay 768 K, interface disco, programas en ROM, mensaje nuevo, etc..., pero podemos cargar el programa D-DAY desde MDV y jugar con gráficos. Solamente los que no saben lo que hacen, son capaces de desmontar la Trump Card para usar un programa mal hecho.
Otro programa de tantos que no corren en QLs ampliados, es el Jungle Eddy. Su versión protegida o desprotegida (versiones para 128K, 640K y 896K) tienen el defecto que después de seleccionar en el menu PLAY, al comenzar a jugar el programa pone en marcha los motores de la unidad de disco. Solución : apagar la unidad de disco antes de jugar.

Tengo que aclarar que la mayoria de los programas que no corren en QLs ampliados son versiones muy antiguas o juegos protegidos. Los juegos protegidos han sido en gran parte desprotegidos y modificados para correr en máquinas ampliadas (algunos ahora no corren en máquinas 128K, solamente en máquinas ampliadas).

Ese ensamblador Z-80 podria ser interesante para la libreria. Por lo menos para ver cómo se hace un ensamblador.
El Supercharge es totalmente incompatible con el Toolkit II. El Turbo puede dar algunos problemas con el Toolkit II (según Digital Precision, el Turbo v2.0 es compatible con el Toolkit II y la Qram), mejor usar el Turbo Toolkit en vez del Toolkit II.
Para lanzar el Editor con ese ensamblador Z-80 compilado con el Turbo es mejor con la última versión del The Editor v2.05, pero solamente hay que cargar un Runtime.
Si vas a usar el Toolkit II en tus programas será mejor que uses el Qliberator v3.22, y no tendrás problemas.
La diferencia entre instalar el Turbo Toolkit o su versión runtime es que el compilador Turbo es una parte integral del Turbo Toolkit (estarias vendiendo una parte no autorizada del Turbo).
El Text 87 puede trabajar con tres tipos de ficheros de texto : _T87, _DOC y ASCII puros.

La serie Forth nunca ha corrido el riesgo de desaparecer, pero tenia que saber

si de verdad habia alguien interesado todavía. El Capitulo IX no ha sido escrito aún, pues queria hacer algo medio espectacular y con el verano encima, y ese montón de cartas que estoy recibiendo (junto con las facturas) no he podido preparar nada.

El Forth de 32 bits fue una realidad por poco tiempo en 1986/87, pero al final han terminado por imponerse por mayoria los PCs, y un Forth de 32 bits no puede correr en una máquina de 16 bits. Creo que tenemos PCs hasta principios del siglo XXI, pues parece que nadie quiere cambiar el Standard del mercado de ordenadores personales (hay mucho dinero por medio invertido). Pero vamos a tener un nuevo Forth Standard ANSI, y si uno se cree capaz, se puede hacer su propia versión Forth de 32 bits, que es lo que han hecho mucha gente.

¡ Bienvenido al Club de colaboradores Forth !, Ya somos tres (si lo deseas, me puedes escribir a una de mis direcciones publicadas en números atrasados).

Salvador Merino
Fuengirola (QLave-154)

RATONES (I)

En el boletín número 2 de este año se publicó una carta de Fernando Martínez en la que, entre otras preguntas, pedía algún programa sencillo en basic para manejar el cursor con el ratón de Investrónica o, mejor aún, de Puricorp.

Este tema me ha interesado desde que dispuse del ordenador con el ratón, ya que me resistía a limitar el uso del "cacharrito" a dibujar con un programa concreto, sin duda interesante, pero también con importantes problemas.

Mis primeros intentos resultaron fallidos, pues las rutinas que era capaz de escribir para un joystik, leyendo normalmente las teclas de cursor o de función, no funcionaban con el ratón; así pues llegué a pensar que era imposible detectar las señales del ratón desde el SuperBasic que era lo único de lo que en aquél momento conocía un poco. En esta situación me decidí a escribir a Puricorp solicitando alguna información complementaria a la que acompañaba al ratón en el momento de su compra (es decir ninguna). Creo que esto debió ser aproximadamente hace unos dos años y todavía espero (pacientemente) alguna respuesta. Por todo ello lo que a continuación os explico se debe únicamente a mi propio esfuerzo, a base de muchas horas de probar varios métodos y de perfeccionar lo que he ido encontrando.

La cuestión principal a la hora de escribir rutinas para el ratón desde el SuperBasic y por la que tuve mis primeros problemas es que, de los diferentes métodos disponibles para recibir una entrada desde el teclado, solamente la función KEYROW es capaz de detectar las señales del ratón. Una vez descubierto este punto lo demás debería ser fácil: se trata de escribir algún tipo de bucle que lea el teclado por medio de esta función, que realice a la vez el tipo de cosas que queramos que haga nuestro ratón y del que podamos salir cuando nos interese (generalmente al apretar el botón). Como veremos más adelante el problema práctico que surge al realizar ésto se deriva directamente de la esencia misma del SuperBasic (y de cualquier otro lenguaje interpretado): la lentitud. Si introducimos demasiadas sentencias dentro del bucle, el tiempo entre una lectura del teclado y la siguiente será tan grande que nuestros movimientos de la mano serán más rápidos, produciéndose el desagradable efecto de tener que mover el ratón largas distancias sobre la mesa o bien hacerlo a una velocidad tan lenta que resulte totalmente inútil el uso del "aparatejo". Así pues es necesario tener en cuenta algunas reglas básicas para escribir estas rutinas: el contenido del bucle debe ser lo más breve posible, todo aquello que no sea necesario repetir más de una vez debe situarse antes o después del bucle, es muy importante reducir al mínimo el uso de estructuras condicionales ( IF ... END IF o SELelct ON ... END SELect), ya que consumen gran cantidad de tiempo. A pesar de seguir todas estas limitaciones al pie de la letra este tipo de rutinas siguen adoleciendo de cierta lentitud al estar escritas en Basic, aunque aún nos queda un modo de eliminar el problema que seguramente ya se le habrá ocurrido a más de uno: se trata de compilar el programa con algún compilador de SuperBasic; yo lo he hecho con QLiberator lo que permite cargar las funciones y los

procedimientos como residentes y usarlos como nuevos comandos del SuperBasic. El rendimiento en esta versión compilada es óptimo para mi gusto, por lo que la he inculido en el paquete de programas que envío a la biblioteca de programas junto con este artículo.

Como ya os he dicho el tema del ratón me ha interesado desde hace tiempo y he pensado en él en muchas ocasiones, por lo que, ante la invitación de Fernando Martínez a escribir un programa, no me he limitado a escribir una única rutina sencillita, sino que he intentado ofreceros unas herramientas flexibles y por lo tanto realmente útiles, esperando que no se queden archivadas sin más. El resultado de todo ello son dos procedimientos y dos funciones, cada uno con diferentes parámetros para ampliar su capacidad de adaptarse a distintas necesidades.

Tanto las rutinas como los procedimientos necesitan conocer algunos parámetros básicos de la ventana en la que van a ser utilizados, por lo que es necesario tener cargada previamente la función WIN_VARS de Brian Hurley, que apareció publicada en QL World en Septiembre de 1.987 y que, según creo, está incuida en la biblioteca de programas del Club. En todo caso la he incluido en el paquete de programas para la biblioteca junto con mis rutinas, así como un pequeño boot que se encarrga de cargarlo todo, siempre y cuando no exista ningún job activo en ese momento. También he incluido el QLib_run de QLiberator, imprescindible para la versión compilada.

El primero de los procedimientos se denomina RATC como contracción de Ratón_Caracteres y su listado es el siguiente:

```
10000 DEFine PROCedure RATC (canal,ctl,vel)
10010   LOCal
              xpos%,ypos%,tecla,bucle,ancho%,alto%,teclado,xcar%,ycar%,axpos%,ay
              pos%,v,i
10020   ancho%=PEEK_W(WIN_VARS(canal)+28)
10030   alto%=PEEK_W(WIN_VARS(canal)+30)
10040   xcar%=PEEK_W(WIN_VARS(canal)+38)
10050   ycar%=PEEK_W(WIN_VARS(canal)+40)
10055   IF vel<=0 THEN v=200:ELSE IF vel=1 THEN v=50:ELSE IF vel=2 THEN v=5:ELSE
              v=0
10060   OVER#canal,-1:selctl
10070   xpos%=PEEK_W(WIN_VARS(canal)+34):ypos%=PEEK_W(WIN_VARS(canal)+36)
10075   IF PEEK_W(WIN_VARS(canal)+34)>ancho%-xcar% THEN
              xpos%=0:ypos%=ypos%+ycar%:IF ypos%>alto%-ycar% THEN
              ypos%=alto%-ycar%:PRINT
10080   REPeat bucle
10085     axpos%=xpos%:aypos%=ypos%
10090     BLOCK#canal,xcar%,ycar%,xpos%,ypos%,7
10100     fila
10150     SELect ON tecla
10160       =selt(1):xpos%=xpos%-xcar%:IF xpos%<0:xpos%=0
10170       =selt(0):xpos%=xpos%+xcar%:IF xpos%>ancho%-xcar%:xpos%=ancho%-xcar%
10180       =selt(2):ypos%=ypos%-ycar%:IF ypos%<0:ypos%=0
10190       =selt(3):ypos%=ypos%+ycar%:IF ypos%>alto%-ycar%:ypos%=alto%-ycar%
10200       =selt(4):EXIT bucle
10210     END SELect
10212     FOR i=0 TO v:NEXT i
10215     BLOCK#canal,xcar%,ycar%,axpos%,aypos%,7
10220   END REPeat bucle
10225     BLOCK#canal,xcar%,ycar%,axpos%,aypos%,7
10230   OVER#canal,0
10240   POKE_W(WIN_VARS(canal)+34),xpos%
10250   POKE_W(WIN_VARS(canal)+36),ypos%
10280 END DEFine RATC
```

La sintaxis del comando RATC, como se puede deducir de la línea 10000, es la siguiente

RATC#canal,ctl,vel

donde

canal es el número de canal asociado a la ventana,

ctl es el número de la puerta de control a la que se encuentra

conectado el ratón. El rango teóricamente es de 1 a 2, aunque cualquier otro número se tomará siempre como 2,

vel es la velocidad de movimiento del cursor. El rango teóricamente es de 0 a 3, aunque cualquier número menor que 0 se tomará como 0 y cualquier número mayor que 3 se tomará como 3.

Ninguno de los tres parámetros es opcional; sí lo es en cambio el signo # antes de canal.
La rutina dibuja en la ventana indicada un cursor de forma rectangular, cuya altura y anchura se adaptan automáticamente al tamaño de caracteres vigente para la ventana en cada momento. El cursor es del tipo del de Quill, es decir fijo cuando está quieto y parpadeante al moverse. La salida del "modo ratón" se obtiene pulsando el botón rojo, pudiéndose entonces realizar alguna operación como inserción o borrado de texto para después volver a entrar en "modo ratón" dentro de algún bucle diseñado al efecto. Los movimientos del cursor siguen en todo momento las líneas de texto y las posiciones de los caracteres, siendo los límites de sus movimientos los del tamaño de la ventana. A pesar de todo lo que os expliqué anteriormente sobre la lentitud de la rutina, una vez optimizado el programa y después de compilarlo, me he visto obligado a introducir el parámetro vel que limita la velocidad de los movimientos, ya que en ocasiones se hacen tan rápidos que es imposible manejar el cursor con suficiente precisión, en especial si utilizais un joystik o las teclas de cursor directamente.

Las dos funciones han sido escritas para utilizarse junto con RATC.

```
10290 DEFine FuNction XAT (canal)
10300   LOCal xpos%,xcar%,valor%
10310   xpos%=PEEK_W(WIN_VARS(canal)+34)
10320   xcar%=PEEK_W(WIN_VARS(canal)+38)
10330   valor%=xpos%/xcar%
10340   RETurn valor%
10350 END DEFine XAT

10360 DEFine FuNction YAT (canal)
10370   LOCal ypos%,ycar%,valor%
10380   ypos%=PEEK_W(WIN_VARS(canal)+36)
10390   ycar%=PEEK_W(WIN_VARS(canal)+40)
10400   valor%=ypos%/ycar%
10410   RETurn valor%
10420 END DEFine YAT
```

La sintaxis de las dos funciones es bastante evidente:

```
XAT#canal     y      YAT#canal
```

donde canal tiene el mismo significado que anteriormente.
YAT nos devuelve el número de fila en la que se encuentra el cursor en la ventana asociada a canal, mientras que XAT nos devuelve la posición del cursor dentro de esa fila, por lo que el uso de ambas funciones es claro en sentencias como

```
550 AT YAT#1,XAT#1:PRINT "..."
```

Como se puede observar tampoco en este caso se admite ningún valor de canal por defecto.

Entre los programas que mando a la biblioteca se encuentra uno con el nombre de Raton_demo1 (no he usado el acento en ninguno de los nombres de los ficheros para evitar problemas). Este programita en realidad no sirve para nada útil y toda su finalidad es la de demostrar prácticamente la utilización de RATC, XAT e YAT. Se activa mediante LRUN, después de haber cargado los procedimientos compilados y la función WIN_VARS con el boot. Inmediatamente aparece el cursor en la esquina superior izquierda de la ventana del canal 1; podeis moverlo con el ratón y probar a ver que sale cuando apreteis el botón. Cuando os hayais cansado de llenar de ratones la ventana podeis simplemente interrumpir el programa con el consabido método de pulsar CONTROL-ESPACIO.

El último procedimiento es sin duda el más complejo y a la vez el más flexible.

```
10430 DEFine PROCedure RATG (canal,px,py,tipo,ctl)
```

```
10435   selctl
10440   IF tipo=0 THEN
10445     sover=PEEK(WIN_VARS(canal)+66):OVER#canal,-1
10450     REPeat bucle1
10460       ax=px:ay=py:CIRCLE#canal,px,py,.5
10470       fila
10510       SELect ON tecla
10520         =selt(0):px=px+2
10530         =selt(1):px=px-2
10540         =selt(2):py=py+2
10550         =selt(3):py=py-2
10560         =selt(4):CIRCLE#canal,px,py,.5:POKE(WIN_VARS(canal)+66),sover:RETurn
10570       END SELect
10575       CIRCLE#canal,ax,ay,.5
10580     END REPeat bucle1
10590   ELSE IF tipo=1 THEN
10595       POINT#canal,px,py
10600       REPeat bucle2
10610         ax=px:ay=py
10620         fila
10630         SELect ON tecla
10640           =selt(0):px=px+1
10650           =selt(1):px=px-1
10660           =selt(2):py=py+1
10670           =selt(3):py=py-1
10680           =selt(4):RETurn
10690         END SELect
10700         LINE#canal,ax,ay TO px,py
10710       END REPeat bucle2
10720     ELSE
10730       sover=PEEK(WIN_VARS(canal)+66):papel=PEEK(WIN_VARS(canal)+68):tinta=PEEK(WIN_VARS(canal)+70)
10740       REPeat bucle3
10750         ax=px:ay=py
10760         FILL#canal,1:INK#canal,papel
10770         OVER#canal,0
10780         CIRCLE#canal,px,py,2
10790         FILL#canal,0:INK#canal,tinta:OVER#canal,-1
10800         CIRCLE#canal,px,py,2
10810         fila
10820         SELect ON tecla
10830           =selt(0):px=px+2
10840           =selt(1):px=px-2
10850           =selt(2):py=py+2
10860           =selt(3):py=py-2
10870           =selt(4):CIRCLE#canal,px,py,2:POKE(WIN_VARS(canal)+66),sover:RETurn
10880         END SELect
10890         CIRCLE#canal,ax,ay,2
10900       END REPeat bucle3
10910     END IF
10920   END IF
10930 END DEFine
10940 DEFine PROCedure fila
10950   REPeat teclado
10960     tecla=KEYROW(fil)
10970     IF tecla<>0 THEN RETurn
10980   END REPeat teclado
10990 END DEFine
11000 DEFine PROCedure selctl
11002   DIM selt(4)
11005   IF ctl=1 THEN fil=1:RESTORE 11060:ELSE fil=0:RESTORE 11070
11020   FOR i=0 TO 4
11030     READ c
11040     selt(i)=c
11050   END FOR i
```

```
11060   DATA 16,2,4,128,64
11070   DATA 16,2,1,8,32
11080 END DEFine
```

Su sintaxis es la siguiente

RATG#canal,px,py,tipo,ctl

donde

canal es el número de canal de la ventana,

px y py indican la posición inicial del cursor con respecto a las coordenadas de gráficos,

tipo indica una de las tres modalidades que se explican más adelante,

ctl es el número de la puerta de control a la que se encuentra conectado el ratón, con idéntico rango que en RATC.

RATG es la contracción de Ratón_Gráficos y su finalidad es precisamente la de utilizarlo en programas de dibujo. Su comportamiento es diferente en función del tipo que le hayamos pasado en los parámetros:
- con el tipo 0 el ratón maneja el movimiento de un pequeño cursor en forma de círculo que nos permite posicionarnos en cualquier lugar de la ventana e incluso salirnos de ella y volver a entrar (si salís, no os alejeis mucho que podeis perderos y va totalmente en serio)
- con el tipo 1 el ratón va dibujando una línea por donde pasa del grosor de un pixel y en el color de tinta vigente para la ventana
- con el tipo -1 se mueve un cursor también circular pero de mayor tamaño, que va borrando todo lo que encuentra a su paso, dejándolo del color de papel vigente.

Para salir del "modo ratón" en cualquiera de los tres tipos basta con pulsar el botón rojo. En esta ocasión no he introducido el parámetro vel para limitar la velocidad, puesto que si estudiais el código vereis que en este procedimiento hay muchas más sentencias incluidas dentro de los bucles, en especial en los tipos 0 y -1, con lo que el movimiento se lentifica. Incluso ha sido necesario recurrir a algún truco adicional para mejorar la velocidad y así en los dos tipos mencionados el movimiento se produce de dos en dos pixels, lo que reduce la precisión en un grado tan pequeño que apenas se aprecia a simple vista.

El programa Raton_demo2 permite observar la rutina en acción; se incluyen unas instrucciones de manejo mínimas en el canal 0 y como en el caso de Raton_demo1 su única finalidad es la de hacer una pequeña demostración práctica. Para salir del programa basta con situar el cursor sobre la banda de FIN y pulsar el botón.

Unas últimas recomendaciones si utilizais la versión compilada, cosa que os recomiendo, es imprescindible pasar los parámetros desde variables declaradas con el nombre que aparece en las líneas de código para poder utilizar los valores devueltos. Además con esta versión el número más alto de canal permitido es 15; si necesitais números más altos (cosa muy difícil) es necesario modificar el programa con la utilidad QLib_patch del QLiberator.

Finalmente os diré que he procurado probar las rutinas en todas las condiciones posibles para evitar la existencia de errores, a pesar de ello es posible que detecteis alguno, por lo que si es así o quereis sugerirme alguna mejora o modificación, o bien darme vuestra opinión, os agradecería que me escribierais (mis señas se publicaron ya en el boletín no 3 de este año).

Posdata: Si os habeis fijado en el título, habreis observado un uno en números romanos entre paréntesis. Eso significa que pienso volver a la carga con más "ratones". Estoy preparando nuevas rutinas, pero esta vez en Forth. Hasta la próxima.

José Carlos de Prada
Madrid (QLave 216)

HARDWARE : Q-POWER REGULATOR
DISTRIBUIDOR : SANDY
PRECIO : 20 libras

He comprado el Q-power y lo he instalado en mi QL. La temperatura ha bajado una barbaridad, y me ha sobrado el regulador 7805 y el radiador por aire.
La instalación de la Q-power es sencillisima, solamente tenemos que abrir el QL (¡ Cuidado con no destornillar los dos tornillos que hay debajo de los MDVs !) y destornillar un tornillo. Después sacar todo lo que sobra y colocar en el mismo lugar la tarjeta Q-power (5x2.5 cm) volviendola a atornillar con el tornillo (el regulador de voltaje tiene tres patas que se enchufan en un conector de plástico que tiene el QL).

Mis problemas antes de instalar este maravilloso regulador eran que mi QL se calentaba muchisimo y mi fuente de alimentación se ponia al rojo vivo. Ahora en pleno mes de julio, mi QL ni se calienta y la fuente de alimentación funciona a una temperatura templada (no ardiendo como antes). Durante más de tres horas de funcionamiento, y usandolo a temperaturas superiores a los 30 grados centígrados, no se ha colgado ni una sola vez (antes en verano se colgaba a las 2 horas y estaba ardiendo al rojo).

Tengo que añadir que mis problemas solamente ocurren en algunos QLs, especialmente aquellos que provienen de una Versión JM o anterior, y que no todos los QLs usan el mismo regulador 7805 (hay muchos modelos de fabricantes diferentes).

De todas la soluciones que he tenido el privilegio de conocer, creo que ésta aunque es la más cara, es la mejor. Bajar la temperatura del ordenador y la fuente de alimentación puede alargar la vida de la máquina, y además el QL es uno de los ordenadores con menos averias que se podia encontrar en el mercado (solamente hay dos cosillas que pueden dar la lata, el teclado y los MDVs) y el más fácil de reparar (uno mismo teniendo los repuestos podria repararlo sin ser un manitas).

He observado sabiendo ya como es un regulador 7805 que el QL posee varios, y la Trump Card posee uno que se encuentra exactamente en ese agujero redondo, que sirve para atornillar la placa de aluminio pintada de negra, que a su vez sirve para disipar el calor.

Adjunto un dibujo que muestra el Q-POWER ya instalado dentro del QL.

Salvador Merino
Fuengirola (QLave-154).

PROBLEMAS DE TEMPERATURA

Hace un año compré por correo una expansión de memoria a Miracle Systems. Cuando la conecté e intenté utilizarla me llevé un buen chasco ya que después de unos minutos el ordenador se volvía loco y empezaba a hacer todo tipo de cosas raras. El problema, por lo que habia leido se debía al calentamiento del regulador de tensión por el exceso de carga que supone la memoria suplementaria. Escribí a Miracle pidiendo consejo y ya os conté el caso que me hicieron. Probé varias cosas y al final encontré la solución definitiva a raíz de un artículo publicado en QL World en Febrero de este año. En este artículo se ofrecen las referencias de varios reguladores de tensión para sustituir el original, pero aparecen algunos errores. Por ejemplo se recomienda la utilización de un regulador 78S05 por admitir una carga de 2 amperios; sin embargo según las tablas que he consultado en realidad está por debajo del amperio que admite el regulador original, además me fue imposible localizarlo por las tiendas de electrónica que visité. Por el contrario el regulador LM340T5/7805, que según el artículo alcanza 1,5 amperios, según las mismas tablas admite más de 2,5 amperios y se encuentra con cierta facilidad en el comercio especializado, por lo que me

decidí a instalarlo en mi QL.

El cambio es realmente sencillo y rápido (se tarda mucho más en abrir el propio QL). Cuando tenemos el ordenador con las tripas al aire se reconoce el regulador fácilmente por estar fijado mediante un tornillo al disipador de calor que está justo detrás de los microdrives. En su parte superior puede leerse su referencia: UA 7805 CSP 8437. Se suelta el tornillo con un destornillador normal y corriente, teniendo cuidado de que no se caigan las arandelitas; se extrae el regulador de tensión tirando de él mientras se sujeta el pequeño zócalo blanco en el que están introducidas las tres patillas; se introduce en el zócalo el nuevo regulador doblándo ligeramente las patillas hasta que apoye su parte posterior sobre el disipador de calor; se vuelve a fijar con el tornillo, colocando las arandelitas en su sitio y por último se vuelve a cerrar la tapa del QL. Yo realicé este cambio hace muchos meses y no he vuelto a tener problemas.

En el QL World aparecen varios anuncios de reguladores de tensión para sustituir el original; sus precios suelen rondar las 4 libras y aveces hay que sumar gastos de envío. El regulador que yo tengo instalado me costó 75 pts que es un precio mucho más razonable para un trocito de silicio fabricado en tiradas de cientos de miles en Taywan.

José Carlos de Prada
Madrid (QL 216)

---------------------------------------------------------------------------

ED.: Después de haber probado el Qpower durante cerca de 4 meses creo que la solución de José Carlos de Prada es la más interesante por ser muy barata, pues la diferencia que se obtiene con el Qpower es que se elimina el calor interior del radiador del QL.

PROGRAMA : QLIBERATOR v 3.22
EDITORIAL : LIBERATION SOFTWARE

Qliberator v3.22, o lo que es lo mismo, en busca de más dinero.

He retornado mi v3.12 en disco con un billete de 10 libras, y he recibido un MDV conteniendo la v3.22 y la segunda edición del manual (cuando ví el MDV, me iba a dar un infarto, pero el MDV estaba en buenas condiciones).

Las diferencias entre la v3.12 y la v3.22 son pocas. La opción SAVE ha sido borrada del menu y se ha agregado un nuevo procedimiento QLIB_USE, que sirve para configurar el Qliberator (casi lo mismo que SAVE, pero más potente y versátil).

QLIB_USE load_device, help_device, x_pos, y_pos, "options bits"

La función QLIB_LIST$ retorna los valores por defectos contenidos por QLIB_USE.

La otra gran diferencia de la v3.22, es el único compilador de Superbasic totalmente compatible con el QPTR INTERFACE. En el manual del QPTR se describe como modificar la v3.1 para hacerla compatible parcialmente (el Turbo v2.0 no es totalmente compatible).

No he encontrado nada más extra que no esté presente en la v3.12. ¡ Sorprendidos, Verdad !. Cuando el año pasado en verano compré y probé este compilador, creia que era imposible que una nueva versión superara a la v3.12. Pero al recibir el QPTR, después de leer las instrucciones, vacilé un poco.

El Qliberator es el compilador más compatible con el Superbasic, pero es el más lento. Aún asi, su velocidad es ligeramente inferior (casi igual) al código más rápido generado por el compilador más rápido de un IBM PC a 4.77 MHz, pero el Turbo v2.0 es ligeramente más rápido que ese mismo código corriendo en un PC

compatible a 8 MHz dotado de un 80386.

Creo que los usuarios de QL, que programan en Superbasic, deberian usar los dos compiladores (el Turbo y el Qliberator), pues aunque el Turbo es el más rápido, no sirve para compilar programas como los Españoles protegidos con FLRUN (el código es malisimo, el programador parece sacado del ZX81), o programas escritos por otras personas.

El Qliberator hace pareja con el Toolkit II (o cualquier Toolkit), pero el turbo necesita su propio toolkit (Diital Precision dice que la v2.0 es compatible con el Toolkit II y la Qram), pues existe alguna pequeña incompatibilidad (se puede solucionar fácilmente a medias) con el Toolkit II y el QPTR.

Después de todo habrá cantidad de usuarios comodones que usaran por comodidad el Qliberator, pues en algunas aplicaciones la diferencia de velocidad casi no se nota.

Salvador Merino
Fuengirola (QLave-154).

PROGRAMA : CPMCOPY y IBMCOPY
AUTOR y DISTRIBUIDOR : D. J. Walker,
22 Kimptons Mead,
Potters Bar,
Herts,
EN6 3HZ.
PRECIO : 20 LIBRAS (Los dos).

Aunque estos programas y autor incluido son practicamente desconocidos, los dos programas a diferencia del Super Media Manager (Digital Precision) funcionan a la perfección y son muy faciles de usar por menus.

Aunque el Discover últimamente se está haciendo famoso, estos programas ya estaban a la venta en Febrero 88. También este autor habia escrito y comercializado hace años un programa llamado OPD/QL FILE INTERCHANGE, el cual permitia a los usuarios de QL leer y escribir ficheros en cartucho de MDV en el formato usado por el ICL OPD y el Merlin TONTO.

Como dato a destacar, debo decir que estos programas han sido escritos totalmente en Superbasic y compilados con el QLIBERATOR (naturalmente corren en multitarea y bastante rápido), lo que significa que solamente se han usado los comandos que ofrece la ROM del interface de disco escrita por Tony Tebby.

El CPMCOPY puede ser usado para capturar programas para los emuladores de CPM o simplemente ficheros de texto o datos. Dispone definidos varios formatos que corresponden a los ordenadores ICL PC1 (40 track), ICL PC1 (80 track), SC84, Kemitron, Amstrad PCW8256, ICL DRS20 y Gemini. Pero podemos definir cualquier formato conocido e incluso se nos ofrece utilidades para definirlo.

El IBMCOPY solamente sirve para capturar ficheros de texto o datos (y algo más como veremos más adelante), pero posee la característica de poder traducir los caracteres ASCII que no son Standard, y que son usados para representar caracteres extrangeros.

En resumen, podemos ver el directorio, leer, escribir, borrar ficheros en un disco Alien (3.5" o 5.25") o pasar esos ficheros a formato QL, pero no podemos formatear un disco en formato Alien (es muy posible que esa restricción no exista en las versiones actuales).

Cuando me referia y algo más, era el programa QS DOS que es un programa escrito en Superbasic (y supongo que compilado) que emula al MS-DOS, y comercializa la

firma italiana SPEM (yo no lo tengo, pero debe correr lentisimo y con muchas limitaciones). De todas maneras Digital Precision aún sigue insistiendo en su propaganda que un tal Grey Matter está escribiendo un emulador de MS-DOS.

Según mi opinión, poder leer la información contenida en los discos en formato IBM ya es algo bastante importante ( y a precio tirado comparado con su equivalente Macintosh-IBM). Los usuarios de paquetes Standard como el Xchange no tendrán muchos problemas, será como trabajar en máquinas compatibles.

Salvador Merino
Fuengirola (QLave-154)

LIBRO : LIBRARY of FORTH ROUTINES and UTILITIES
EDITORIAL : SHADOW LAWN PRESS
AUTOR : James Terry
PRECIO : 35 dólares con transporte aereo desde California.
IDIOMA : Inglés
PAGINAS : 374

La única manera que conozco para poder comprar este libro, es ser socio de FIG (California). Si alguien está interesado en hacerse socio, podria pedirme a mi una solicitud, pues si puedo hacer un socio nuevo, al ser socio antiguo, me regalan un libro.

Antes de comentar el libro voy a andar un poquillo por las ramas (seria interesante que los socios conozcan la actualidad).
Se está cociendo el nuevo Forth Standard ANSI 1988, y por compatibilidad con todos los sistemas se mantiene a 16 bits el Stack y las direcciones, pero existiran palabras de diferente tamaño.
Este nuevo Forth nace para conseguir por fin un Forth Standard de verdad, y no la cantidad de dialectos que existen actualmente (79, 83, FIG, etc..., y dentro de esos dialectos más sub-dialectos)
Aún asi, hay usuarios que no están conforme, y usan nuevos sistemas Forth no standard de 32 bits de direcciones y stack de 16 bits o 32 bits. Y por lo que veo en la tercera reunión para el nuevo Forth ANSI, han ganado los que querian 16 bits de stack y direcciones (aunque desde luego habia una gran presión por los fabricantes, vendedores y diseñadores, pues un Forth de 32 bits solamente puede correr en una máquina de 32 bits o más, y seria incompatible con los programas actuales. P.e.: un PC solamente podria usar un Forth de 32 bits de direcciones si usaba un 80386). Ya explicaré como acceder a más de 64 K, pues aunque el QL no usa el clásico sistema del baile de los segmentos, puede utilizar cualquier dirección disponible en su mapa de memoria (con cuidado de no hacer un Crash del sistema).

Según el libro, el autor habia comenzado a interesarse por el Forth cuando vió en TV un episodio de "Fame" (Fama) en 1983 (se usaba un ordenador programado en Forth para un montaje espectacular).

Actualmente el Forth es uno de los lenguajes favoritos usados en Inteligencia Artificial y sistemas expertos.

Este libro se divide en 13 capitulos, que son: 1- Introduction, 2- Case Statements, 3- A programmer's calculator, 4- full-screen editor, 5- 8088 Macro assembler, 6- 8087 Numerical Coprocessor, 7- Strings, 8- Input formatting, 9- Displays and Output formatting, 10- Natural Language processing, 11- Data Structures, 12- Expert systems, y 13- Debugging programs (he usado los titulos originales, pero no creo que nadie tenga dificultades a estas alturas).

En la introducción se explica que el dialecto Forth usado es el Atila, y el ordenador IBM PC y compatibles, pero que las palabras usadas pueden usarse en cualquier sistema Forth. El Atila está disponible para el Apple Macintosh, Atari ST y Amiga desde 1986 (Año en el que Alan Sugar asesinó al QL).
El Atila posee la caracteristica de que aunque 16 bits solamente permite 64 K de

direcciones, puede utilizar los 640 K de un IBM PC usando punteros de 32 bits. Este sistema consiste en usar varios segmentos de 64 K que permiten poder acceder a la Screen del IBM PC, tener el diccionario en un segmento y los datos, textos, gráficos repartidos en diferentes segmentos. También se explican con detalles las palabras disponibles en el Atila (para usar la memoria extra) para que el usuario busque las equivalentes en su manual (estas palabras no son standard, y se usan en los capitulos 4 y 9).

El Superforth es un sistema Forth-83 que corre como un Job. Esto permite lanzar varios Superforth a la vez, pero ¿ Cómo usar más de 64 K ?.

Por lo pronto, el Superforth usa 68 K, en vez de 64 K. ¡ Increible, verdad !. La razón es muy sencilla, una parte del código y mensajes de error están situados fuera del diccionario para que el usuario disponga de más espacio.
Si leemos el contenido de 3.12 FURTHER MEMORY HANDLING , observaremos que el Superforth ha sido dotado de un juego de palabras que puede poner o coger ( alterar) el contenido de cualquier posición de la memoria del QL. Estas palabras trabajan con bytes, palabras y palabras largas. Con ellas podemos acceder a la Screen y otros dispositivos. Pero ¿ Cómo usarlas sin hacer un Crash del sistema ?.

Con la ayuda del QDOS podemos reservar un espacio de la memoria del QL, y usarla para guardar datos de cualquier tipo (Arrays, gráficos, etc....). Hay programas que usan esa opción. P.e.: el editor de metacomco cuando pregunta "¿ Cuánto espacio de datos ?". El Vroom que es un Job, cuando carga los gráficos que están en otro fichero en memoria desde dentro del programa.
Después de todo, el corazón de casi todos los programas puede caber en 64 K (o 55.000 bytes libres) y si no cabe, cosa rara (el código del Superforth es muy compacto y usa muy poca memoria), siempre podriamos dividir el programa en pequeños Jobs que controlados por el job principal, darian el mismo resultado que el programa grande, pero seguramente seria más versátil (tendria opciones fuera de lo normal).

Los capitulos 4, 5, 6 y 9 solamente sirven para el IBM PC. Pero ayudan muchisimo a observar desde el punto de vista de la programación cómo funciona un IBM PC, y apreciar aún más nuestro QL y su maravilloso QDOS. Nuestro sistema operativo es muy superior al MS-DOS, pero nos hace falta más Software. Hace tiempo que un socio de Quanta comprobó que el paquete de coma flotante del QL a pesar de todas sus limitaciones, era más rápido que un 8088 con un coprocesador númerico 8087 (este último ofrece unas posibilidades de cálculo que nuestro paquete de coma flotante no ofrece, pero para eso está el MC 68020 y MC 68881).
El capitulo de las cadenas es muy interesante, pero la libreria de palabras para usar cadenas del Superforth es muy completa (tan completa como nuestro QDOS).
El capitulo Input formating, aunque es interesante (siempre es bueno conocer varios estilos, el Superforth tiene un surtido de palabras para hacer eso.
El capitulo Natural Language processing forma parte de una de tantas posibilidades que ofrece el Forth, y yo aún casi desconozco. Este capitulo sirve para poder comunicar ordenes en lenguaje normal (perfecto Inglés), pero creo que se podria hacer una versión es castellano a estilo tarzán (yo ser Tarzán, tú ser Chita).
El capitulo Data Structures me permitira crear nuevas estructuras de datos a mi medida. Se explica con detalle como usar la memoria virtual en disco, crear matrices, matrices en disco, almacenamiento de datos por campos en un record, etc..
El capitulo Expert Systems puede ser el más interesante de todos junto Data Structures. Se explica con detalle todas las definiciones de las nuevas palabras para crear nuestro sistema experto. Os puedo adelantar que os podeis reir de ese programa llamado QL-EXPERT si envio a la libreria una versión de esta libreria de palabras para usar con el Superforth (según el libro, el usuario puede usar en sus programas todas las palabras del libro). No voy a explicar nada sobre sistemas expertos porque lo único que sé fue lo que me explicaron en filosofia en tercero de BUP hace 8 años, pero no parece muy dificil pues todo es lógica.
El capitulo Debugging Programs ofrece unas cuantas palabras para hacer más fácil la depuración de los programas (en busca de errores). Depurar las palabras en Forth es bastante fácil debido a que es un lenguaje interpretado y compilado a la vez.

En el apendice podremos encontrar las palabras que se suponen que pueden ser especificas Atila (es un dialecto del Forth 79). Estas palabras son :

```
: ENDIF COMPILE THEN ; INMEDIATE
: <BUILDS COMPILE CREATE ;
: LROT SWAP ROT SWAP ;
: HOME CLS ;
: ATILA FORTH ;
```

Las palabras VTAB y HTAB están definidas en código máquina 8088, pero equivalen a AT.
Existen muchas palabras que no son Forth 79 Standard (y menos aún Forth-83 Standard), pero las conozco todas porque están explicadas y definidas en el libro Lenguaje Forth para micros (Paraninfo).

Tengo que destacar que para programar en Forth, no hace falta un IBM PC o compatible, y el QL es una joya preciosa que no tiene nada que envidiarle.

Salvador Merino.
Fuengirola (QLave-154).

LIBRO : FORTH. Anatomía de un lenguaje inteligente.
AUTOR : Ferrer Abelló, Antonio M.
EDITORIAL: INGELEK S.A.
PVP : 395 pts.

Este es uno de los tres libros que he utilizado junto con muchas revistas para empezar a introducirme en el manejo del FORTH. Se trata de un pequeño volumen (103 páginas), incluido en la colección Biblioteca Básica de Informática que la editorial Ingelek lanzó en los kioscos de prensa hace un par de años y que todavía hoy puede localizarse en librerías especializadas. Su precio es muy asequible comparándolo con el del reto de libros, especialmente si se trata de libros de informática. Otra característica a resaltar es la de ser el único (al menos que yo conozca) escrito originalmente en castellano sobre este lenguaje; el resto de bibliografía en nuestro idioma sobre el FORTH se reduce al volumen de Steve Oakey publicado por Paraninfo y que comento más adelante.

Pasando al contenido hay que decir que se tocan prácticamente todos los puntos tradicionales en lo referente a este lenguaje: la palabra, las pilas, las operaciones básicas, creación de palabras, manejo de la memoria, variables y constantes, estructuras de control. Como es fácil de comprender un programa tan amplio en un espacio tan reducido determina un tratamiento superficial de todos los temas. A pesar de ello el libro es claro y sencillo, siendo quizás estos aspectos de brevedad y sencillez juntos los que lo hagan muy indicado para un primer acercamiento.

Junto con sus virtudes ya señaladas hay que apuntar algunos fallos: el capítulo dedicado a variables, constantes y tablas, a pesar de ser claro en la exposición general, deja en completa oscuridad el tema de la creación de nuevas palabras por medio de "palabras definitorias" (CREATE ... DOES>). Otro capítulo a todas luces insuficiente es el dedicado a las matemáticas en coma fija, que son tratadas por el autor como un defecto del FORTH y no como una de sus virtudes, lo que denota falta de profundización en la filosofía tan especial de este lenguaje.

Se incluye el juego de instrucciones del FORTH-79, alegando que éste se encuentra mucho más difundido que el FORTH-83, lo cual pudo ser cierto en el momento de escribirse el libro pero no ciertamente hoy en día.

Una vez dicho todo esto, sigo recomendando este libro para cualquiera que no haya tenido contacto anterior con el FORTH y sienta alguna curiosidad por él, ya sea con la intención de continuar después adentrándose por este camino o simplemente para tener un conocimiento somero.

José Carlos de Prada
Madrid (QLave-216)

LIBRO : STARTING FORTH
AUTOR : Leo Brodie y Forth, Inc.
EDITORIAL : Prentice Hall
PVP : depende

"Dulcis in fundo", decían los romanos, así que he dejado para el final la perla de mis libros sobre informática, la "Biblia del FORTH", como lo llaman algunos.

Para comenzaros contaré una pequeña historia que aclara la cuestión del precio. Hace algo más de un año me encontraba yo con mi cartucho de FORTH Computer One recién comprado y sin un buen libro introductorio que llevarme a la boca. Decidí acercarme a una librería especializada de renonmbre (Díaz de Santos). Allí efectivamente me mostraron un pequeño rincón dedicado al FORTH (30 o 40 cms de estantería): el único libro en castellano era el de Steve Oakey que ya poseía y sobre el que ya conoceis mi opinión. Con respecto a los libros en inglés y tras una somera ojeada, no encontré ninguno que me convenciera lo suficiente para compensarme de los precios astronómicos que tenían. Pregunté por el libro de Leo Brodie del que ya había oido hablar; me dijeron que en ese momento lo tenían agotado en la librería, pero que me lo podían pedir a Estados Unidos sin ninún problema. Antes de tomar la decisión quise saber aproximadamente el precio, a lo que me respondieron sin darle mayor importancia que no pasaría de las 8.000 pts. Lógicamente decidí pensármelo más despacio.
Al cabo de unos meses ingresé en el Forth Interest Group, lo que de por si me ha ayudado mucho a profundizar en el FORTH. Una de las ventajas que ofrece la pertenencia a esta organización es precisamente la de tener acceso a una gran cantidad de bibliografía sobre el lenguaje con cierto descuento. Mi sorpresa (agradable desde luego) fue la de encontrar en la lista de libros disponibles el Starting Forth 2a edición al precio de 21$, por correo de superficie o 30$, por correo aéreo. Me parece que por mucho descuento que nos ofrezca el FIG a sus miembros, la diferencia de 3000 a 8000 pts. ofrece sin duda un buen margen comercial para cualquier librero. Por si alguien está interesado en intentar pedir el libro directamente a la editorial su dirección es la siguiente:

PRENTICE-HALL, INC.
Englewood Cliffs, New Jersey 07632
USA

Dicho todo esto, volvamos al contenido del libro, que adecir verdad vale su precio en oro. El volumen tiene 346 páginas, lo que da idea de un mayor detenimiento en cad uno de los temas. Lo primero que encontramos es el prólogo, escrito nada más y nada menos que por Charles H. Moore, el creador del FORTH, y que comienza de con las siguientes palabras: "La comunidad Forth puede celebrar un evento significativo con la publicación de Starting Forth. Un mayor esfuerzo, talento y empuje se han puesto en este libro que en cualquier manual introductorio anterior."

El estilo de Leo Brodie es fácil, brillante, atractivo, con constantes toques humorísticos. El libro está ilustrado por el mismo Brodie con personajes que, como señala Moore, hacen el estudio fácil y divertido.

Se revisan a lo largo de sus páginas todas las cuestiones básicas y de fondo referentes al lenguaje, notándose epecialmente la creatividad, conocimientos y capacidad pedagógica del autor en los temas que normalmente resultan más áridos y que aquí se convierten en un pasatiempo apasionante. Son muy de señalar los capítulos dedicados a la obtención de resultados, a las características especiales de las estructuras de control en el FORTH, a la aritmética entera, a los sistemas de numeración, a la estructura interna del lenguaje y a las palabras para definición y compilación.

A todo esto hay que añadir un estilo de programación en todos los ejemplos muy cuidado, que resulta asombroso por su simplicidad y que poco a poco se te va pegando casi sin que te des cuenta. Al final de cada capítulo hay una serie de problemas, cuyas soluciones se encuentran al final del libro. Tanto estas soluciones como los ejemplos diseminados por todas sus páginas son en muchas ocasiones, además de una útil guía para aprender, herramientas muy válidas que se pueden incluir en cualquier implantación FORTH.

Al final sólo un defecto: como ya habreis podido deducir el libro no está aún traducido al castellano, aunque pienso que no resultará excesivamente difícil seguirlo para cualquiera que posea unos conocimientos mínimos de inglés.

Starting Forth es sin duda un libro altamente recomendable para cualquiera que tenga interés por el FORTH, ya sea un completo neófito o posea conocimientos previos sobre él y si no que se lo pregunten a Charles Moore, que reconoce haber disfrutado de su lectura a pesar de ser el único en el mundo que en ningún momento a necesitado aprender FORTH.

José Carlos de Prada
Madrid (QLave-216)

LIBRO : Lenguaje FORTH para micros.
AUTOR : Steve Oakey
TITULO ORIGINAL : FORTH FOR MICROS
EDITORIAL : PARANINFO
PVP : 1.350 pts.

El libro es algo más extenso que el de Ferrer Abelló, a cambio es mucho más oscuro y farragoso, no se si debido al original en inglés que no conozco o a la traducción, que desde luego no es muy brillante.

El temario es muy similar al anterior, aunque el tratamiento es muy diferente. Los temas se pretenden desarrollar con mayor profundidad, aunque con diferente fortuna y deteniéndose a veces en aspectos más anecdóticos que sustanciales.

Uno de los aciertos del libro es la introducción al tema de la "notación polaca inversa", que suele ponerse cuesta arriba para el neofito y que aquí resulta sencilla casi desde el principio.

De nuevo el tema de la aritmética en coma fija se aborda de pasada y sin convicción, abogándose más bien por las extensiones para tratamiento en coma flotante.

En el libro se incluyen bastantes ejemplos prácticos de programación, extendiéndose muy especialmente en la implantación del método del Vector Dope, sin que a fin de cuentas se sepa muy bien porqué, ya que no aporta mucho desde el punto de vista pedagógico y en la práctica existen soluciones mejores y más simples para el mismo problema. En general cabe señalar que la mayoría de los ejemplos se alejan bastante del estilo característico del FORTH: se nos presentan palabras de hasta casi 30 líneas, mientras Charles Moore (creador del FORTH) se muestra cada vez más partidario de que las palabras FORTH no superen las 2 o 3 líneas; se omiten por completo los comentarios de efectos sobre la pila, que son un elemento básico para la legibilidad de los programas y no se aclara suficientemente la factorización de funciones a la hora de programar.

Al final del libro se incluye un capítulo dedicado a las distintas versiones del FORTH y a facilitar algunas conversiones entre ellas, lo que pude ser útil de cara a comprender programas escritos para viejos sistemas todavía muy en uso como FIG-FORTH, MMS-FORTH y otras, aunque no se incluye el estándar del 83 por la fecha del original, que es el más utilizado actualmente y en el que se basan las dos implementaciones disponibles para el QL.
En resumen, se trata de un libro al que es difícil hincarle el diente, confuso en muchas ocasiones y no muy recomendable para iniciarse, pero del que se pueden extraer algunas ideas concretas si antes se manejan los conceptos básicos.

José Carlos de Prada
Madrid (QLave-216)

LIBRO : LENGUAJE FORTH PARA MICROS

AUTOR : STEVE OAKEY
TITULO ORIGINAL : FORTH FOR MICROS
EDITORIAL : PARANINFO SA, Magallanes 25, 28015-MADRID
PVP : 850 # (cuidado, precio especial del Corte Inglés)

FORTH es un lenguaje relativamente nuevo que está teniendo una gran acogida por parte de los usuarios de microordenadores: es más fácil de usar y más rápido que el BASIC. El Forth es un lenguaje "ensamblador de alto nivel", con todas las ventajas de un ensamblador y ninguno de sus dolores de cabeza, siendo más fácil de aprender y utilizar. FORTH se asemeja a un intérprete, con todas las ventajas que ello comporta: fácil desarrollo de programas y fácil depuración de errores. FORTH es un lenguaje en desarrollo y esto crea un problema menor: no tiene un repertorio único estándar de instrucciones.
Este libro tiene un interés particular para los entusiastas del micro y los estudiantes de microinformática ( para mi, es una pequeña joya para mi colección de libros).
Con la ayuda de los test de autoevaluación, usted podrá aprender a programar en FORTH y a comprender el uso y las ventajas de este lenguaje.
FORTH está disponible para una gama muy amplia de micros: ZX Spectrum, ZX81, BBC, Dragón 32, Oric, Acorn Atom, TRS80, Apple, Pet, Jupiter ACE, Commodore 64, etc (naturalmente, también en los 68.000 y PCs), y cada vez tiene mayor popularidad.

¿ Qué va a encontrar en el libro ?

1. INTRODUCCION AL FORTH. ¿Qué es el Forth?. ¿Qué se asume en este texto?
2. ARITMETICA. Notación Polaca invertida. La pila. Limitaciones aritméticas. Tratamiento de las pilas. Ejercicios.
3. PROGRAMAS. La palabra. Cómo usar parámetros a las subrutinas. Utilización de parámetros múltiples. Obtención de los resultados de las subrutinas. Obtención de los resultados múltiples de las subrutinas. El diccionario. La pila de retornos. Ejercicios.
4. SELECCION. Booleanos. Operadores Booleanos. La sentencia IF. Ejercicios.
5. REPETICION. El bucle sencillo DO. El bucle DO extendido. Bucles DO anidados. El bucle WHILE. El bucle REPEAT. El bucle WHILE generalizado. Ejercicios.
6. ENTRADA NUMERICA Y ARITMETICA AMPLIADA. Palabras básicas de entrada. Entrada numérica. Números sin signo. Números de doble longitud. Operadores para distintas longitudes. Enteros de distinta longitud y coma flotante. Operadores aritméticos y especiales. Bases de numeración. Utilización de CONVERT. Ejercicios.
7. TIPOS DE DATOS. Caracteres. Variables. Constantes. Arrays. Cadenas. Ejercicios.
8. ENTRADA/SALIDA EXTENDIDA. Salida formateada. El buffer de entrada y WORD. Pantallas. Otras salidas. Ejercicios.
9. AMPLIANDO EL FORTH. Declaración de estructuras de datos simples. Declaración de nuevos tipos de datos. Implantación de estructuras de datos complejas. Compilación frente ejecución. Recursividad.
10. DESARROLLO DE PROGRAMAS. El proceso de diseño. Codificación. Prueba del programa. Arrays multidimensionales. Implantación del método Vector Dope.
11. VERSIONES DE FORTH. fig-FORTH. El FORTH del Dragón. El Júpiter Ace.
En el apéndice 3. Respuestas a los ejercicios.

Ojo, este libro ha sido editado en 1985, y lo he encontrado en la limpieza que ha hecho el Corte Inglés en su almacén en Málaga. Por eso el PVP es tan bajo, porque habian perdido el precio. El libro tiene 176 páginas.
Este libro no pretende definir las palabras FORTH 79 Standard (ayuda solamente a usar las más comunes), sino intentar que el lector aprenda a programar en FORTH. Los ejemplos en el libro son muy numerosos y el mismo ejemplo viene en PASCAL, BASIC y FORTH, para que el usuario que proviene del Pascal o el Basic comprenda mejor como funciona el FORTH.

Ya sabeis que el libro hace años que fue editado y es dificil conseguir una copia (podeis intentarlo en el Corte Inglés). De todas maneras, creo que es un libro muy interesante y de fácil lectura. Lo que no comprendo es por qué no fue nunca comentado en una de esas revistas de informáticas de nuestro querido pais.

SALVADOR MERINO
FUENGIROLA (QLave-154).

## FORTH

### COMANDOS PARA DIBUJO EN TRES DIMENSIONES (II)

Para completar mi colaboración anterior sobre el mismo tema paso a mostraros una nueva palabra Forth, que en este caso sirve para dibujar en el espacio definido por tres ejes de coordenadas x,y,z un círculo, teniendo en cuenta la distancia del observador y la inclinación con respecto a él en dos sentidos diferentes, con lo que se puede lograr cualquier posición en el espacio.

```
: CIRCULO3D  ( x y z radio rotación giro -- )
             5 ROLL FLOAT 7 ROLL FLOAT 9 ROLL FLOAT 11 ROLL FLOAT
             13 ROLL FLOAT 15 ROLL FLOAT
             5 FPICK F# 100 F* 4 FPICK F/ F# 100 F+  ( x en 3d )
             5 FPICK F# 100 F* 5 FPICK F/ F#  50 F+  ( y en 3d )
             4 FROLL F# 100 F* 5 FROLL F/              ( radio )
             6 FROLL F# 2   F/ 4 FPICK RAD FCOS  F*
             6 FROLL F# 2   F/ 5 FPICK RAD FSIN  F*
             6 FROLL F+ F+  RAD  FCOS         ( excentricidad )
             4 FROLL                                 ( ángulo )
             ELLIPSE   ;
```

Como podeis ver CIRCULO3D espera disponer de seis parámetros situados en la pila en el siguiente orden:
x, y, z, que son las coordenadas del centro del círculo con respecto a los tres ejes y deben cumplir los requsitos establecidos en las palabras anteriores.
radio es lógicamente la medida del radio del círculo.
rotación se refiere al ángulo de rotación del plano del círculo entorno a su propio diámetro.
giro es a su vez el ángulo de rotación del mismo plano con respecto a un diámetro perpendicular al anterior

Los dos últimos parámetros deben ser introducidos en grados y para su posterior transformación en radianes se ha utilizado la palabra RAD que debe estar cargada con anterioridad.

```
: RAD   ( fgrados -- fradianes )
        F# 6.283184 F* F# 360 F/ ;
```

Las palabras mostradas hasta aquí deben considerarse como meros puntos de partida para posteriores aplicaciones. La belleza del Forth está precisamente en su capacidad para expandirse sin límites. Para dar simplemente una idea de todo ello os planteo una sugerencia de como continuar desarrollando el dibujo en tres dimensiones en forma de pseudocódigo adaptado al Forth:

```
ESFERA
         Inicio
                   Pila  <-  ( x y z radio )
                   Desde I=0 hasta 330  Incremento de 30
                             Duplicar ( x y z radio )
                             Pila  <-  ( I 0 )
                             CIRCULO3D
                   Fin_Desde
         Fin
```

El transformar este algoritmo en una palabra Forth es algo casi directo que os dejo a vosotros como ejercicio, así como el planteamiento de nuevos algoritmos para dibujar cubos, cilindros, etc.

José Carlos de Prada
Madrid (QLave 216)

PRACTICANDO CON EL SUPERFORTH (VIII)

Ya es Mayo (mi cumpleaños es el dia cuatro, 24 añitos.). Como no tengo ganas de usar mucho este trasto negro en verano, voy a escribir lo que se pueda en varios ratos libres, pues solamente me interesa conocer los nuevos bañadores que van a usar las chicas este verano, y descansar todo lo que pueda en las horas libres fuera del horario de trabajo.

Si alguien interesado por el FORTH ha podido hacerse con el libro LENGUAJE FORTH PARA MICROS, se habrá dado cuenta que no he inventado nada nuevo, y que esta serie podria haberse suprimido. Pero da la casualidad que los Españoles compran muy pocos libros y rara vez los leen. Por eso, ese libro ha pasado sin pena y gloria por la libreria.

En los últimos capitulos he metido más de una vez la patita, pues se me ha olvidado algunos pequeños detalles. Pero si alguien tiene alguna duda, escriba a la sección preguntas, yo intentaré responder a su pregunta (en otra sección con el nombre de RESPUESTAS).

En este capitulo voy a tocar la RECURSIVIDAD en plan cateto y resumido.

La recursividad se usa mucho en C y LISP. Es otra forma de implantar un bucle de una forma corta y muy fácil de entender. Pero mejor ver un par de ejemplos clásicos :

```
: FACTORIAL
   DUP 0= IF              ( Si se quiere factorial de 0 )
    DROP 1                ( Devuelve el valor 1 )
    ELSE
    DUP 1-                ( Guarda una copia de N, y forma N-1 )
    RECURSE               ( Calcula Factorial [N-1] )
    *                     ( y la multiplica por la copia guardada de N )
    THEN   ;
```

```
5 FACTORIAL .    Imprimirá 120
```

Como no se puede usar el nombre de la palabra como parte de su propia definición, el SUPERFORTH ofrece la palabra RECURSE. La anterior definición lo que hace es ir formando N-1 e introduciendolo en la pila. Al final cuando llega a cero, se sale de la definición principal y entra en acción el * tantas veces como ha sido usada la palabra RECURSE. El resultado es un número de 16 bit siempre, y si es superior a 16 bit dará un resultado erróneo.

El siguiente ejemplo es casi análogo al anterior (hemos cambiado * por . ).

```
: LISTA_ASCENDENTE DUP 0> IF DUP 1- RECURSE THEN . ;

5 LISTA_ASCENDENTE     imprimirá   0 1 2 3 4 5
```

Otro ejemplo de factorial, pero con números en coma flotante.

```
: FACTORIAL FDUP F0= IF FDROP 1 S->F ELSE FDUP  1 S->F F- RECURSE F* THEN ;
```

Como podrán observar, el único defecto que van a encontrar es la baja precisión que posee el paquete de coma flotante suministrado en la ROM del QL Este defecto fue duramente criticado durante los primeros meses de vida del QL hasta su muerte (la compra de los derechos por Amstrad), pero si alguien necesita más precisión podria encontrarla en los compiladores Lattice C (Metacomco) y Pro-Fortran 77 (Prospero), que creo que tienen 16 digitos de precisión (Aunque una solución podria ser, escribir un nuevo paquete de coma flotante). Como factorial de 10 da un resultado de 3.6288E6.

Como creo que más de uno se habrá dado cuenta, la recursividad es un método sencillo y elegante que ahorra mucha memoria en un sistema FORTH.

Aunque no tenia pensado hacerlo, creo que con un poco de suerte puedo volver a colocar mi pequeño rincón dedicado al ASSEMBLER.
La primera palabra que voy a definir es una versión a mi manera de la definición de factorial en ASSEMBLER (usando el ensamblador de Metacomco).

```
* Por aqui arriba van los macros (ver capitulo VII)
******************************************
*
               code     'FACTORIAL',factor_end
               tst.w    d2              comprobar si TOS es cero
               beq      cero            si lo es salta a cero
               move.w   d2,d1           haz una copia de TOS en d1
bucle
               move.w   d2,-(a3)        guarda una copia de TOS en 2OS
               subq     #1,d2           forma N-1
               tst.w    d2
               bne      bucle           el bucle se repetirá hasta que TOS sea
cero
               move.w   (a3)+,d2        hacemos un DROP
               subq     #2,d1           el nuevo bucle se repetirá N-1 veces
calcular
               mulu     (a3)+,d2    multiplicamos 2OS por TOS. Resultado en TOS
               dbf      d1,calcular
               bra.s    hecho
cero
               moveq    #1,d2

hecho
               next


factor_end
               end_file

               end
```

Una vez que hemos cargado la utilidad del block 8, podemos cargar fichero objeto de la palabra anterior con LOAD_BIN. Si tecleamos 5 FACTORIAL . (ENTER), el resultado será 120.

Esta definición de factorial solamente calcula hasta factorial de 8. Al pasar de ese limite los resultados son incorrectos, pero si deseamos un resultado 0 que indique error, tendremos que añadir al programita anterior unas lineas más en el lugar adecuado. P.e.:

```
         cmp.l     #65535,d2
         bhi      error

         ...       ....

error    clr.l     d2
         bra.s     hecho
```

La siguiente definición en ASSEMBLER es otra versión de factorial, pero que permite un resultado de 32 bit. Por lo tanto, sirve para calcular funciones factoriales más grandes, pero no se hagan muchas ilusiones .

```
* En algún lugar debe haberse escrito antes los macros (ver capitulo VII)
******************************************
*
               code     'FACTORIAL',factor_end
               tst.w    d2
               beq      cero
               move.w   d2,d1
```

```
bucle
               move.w   d2,-(a3)
               subq.w   #1,d2
               tst.w    d2
               bne      bucle
               subq.w   #2,d1         el bucle calcular se repetirá N-1 veces
               move.w   (a3)+,d3      sacamos 2OS de la pila y lo ponemos en d3
               ext.l    d3            si no se extiende el signo, obtendremos
                                      un resultado incorrecto.
               move.l   d3,d4         d4 igual a d3
calcular
               move.w   (a3)+,d2      sacamos 2OS de la pila y lo ponemos en d2
               subq.w   #2,d2         el bucle2 se repetirá las veces que
                                      contenga d2 - 1

bucle2
               add.l    d3,d4         suma d3 a d4. Resultado en d4
               dbf      d2,bucle2     si d2 no es cero, se repite bucle2
               move.l   d4,d3         d3 igual a d4
               dbf      d1,calcular   si d1 no es cero, se repite calcular
               move.w   d4,-(a3)
               swap     d4
               move.w   d4,d2
               bra.s    hecho
cero
               move.w   #1,-(a3)
               moveq    #0,d2

hecho
               next

factor_end
               end_file

               end
```

Si tecleamos (se supone que hemos cargado la nueva palabra) 19 FACTORIAL D. , el resulatdo será 109641728. En esta palabra si nos pasamos del limite, lo único que puede pasar es lo mismo que en el contador de KM de un coche. Para detectar ese error y perdida de tiempo, lo único que tenemos que hacer es detectar el rebose y saltar a la etiqueta error.

Es muy posible que os esteis preguntando si es posible hacer que el resultado sea un número de 64 bit. Os puedo adelantar que en código máquina es fácil, pues está previsto. Lo que no están hechas son las rutinas de conversión de 64 bit en binario  a decimal y viceversa (y otras zarandajas). De todas maneras antes de meterle mano a ese asuntillo, prefiero echarle un vistazo a un algoritmo que utiliza un programa calculadora de QUANTA que usa una precisión de 120 digitos (con suma, resta, multiplicación y división). Según tengo entendido el programa está escrito en Superbasic y es lentisimo (pero funciona).

Los métodos para llevar a cabo el objetivo anterior son:

- Utilizar las instrucciones aritméticas   extendidas que ofrece nuestro superpotente ensamblador 68.000. En este caso tendremos que sudar la gota gorda escribiendo las  nuevas  rutinas  para paso  de ASCII a enteros binarios  y viceversa. Para más detalles aconsejo que lean el libro Matemáticas para Programadores de ANAYA MULTIMEDIA, que aunque ha sido escrito pensando en los pequeños ordenadores de 8 bit es muy bueno para los principiantes de código máquina.

- Utilizar el BCD (decimal codificado en binario).   Aunque esta solución tiene un coste en memoria y velocidad, es la solución más fácil e interesante.

Es muy posible que me decida por escribir las dos versiones definiendo las nuevas palabras en Superforth y Assembler. Lo que aún no sé, es cuando lo voy a hacer, pues ahora estoy en una época  de crisis.

Salvador Merino
Fuengirola (QLave-154).

CONTROL DE VENTANAS DESDE EL FORTH-83

A través de los boletines de QLave he podido comprobar un creciente interés por el FORTH entre los usuarios españoles del QL, a pesar de que este lenguaje no esté muy difundido en nuestro país para otros ordenadores. Desde el año pasado soy miembro del FIG(Forth Interest Group) y debo recomendar esta asociación para todo aquel que se encuentre intersado por este lenguaje tan especial.
La implementación que poseo es la de "Computer One" y tengo que decir que en general me encuentro muy satisfecho con sus prestaciones, aunque después de leer el comentario de Salvador Merino en el número de Octubre sobre el SuperForth de Digital Precision, estoy deseando conocer esta versión. En todo caso, con respecto a la crítica en ese comentario sobre la aparición del copyright en los programas escritos con el Forth-83 de Computer One, hay que aclarar que en esta versión se dispone de palabras adecuadas para manejar la mayoría de las posibilidades del QDOS, por lo que resulta muy fácil modificar las ventanas para la presentación de cualquier programa. La misma cuestión se puede plantear en términos de SuperBasic: si no utilizamos las facilidades para el manejo de las ventanas, tendremos que presentar todos los programas en una pantalla dividida en tres zonas diferentes, que es como aparece la pantalla después de pulsar F1, pero es que ¡para eso están todos los comandos referidos a las ventanas!
Ya que el movimiento se demuestra andando, os propongo como ejemplo una pantalla que utilizo en un programa de dibujo en tres dimensiones que estoy desarrollando.

```
FCB DIBUJO
DIBUJO FILENAME SCR_512x196a0x0
DIBUJO OPEN-FILE DIBUJO CHANNEL IS-WORK BLACK PAPER CLS
3 WHITE BORDER

FCB MENU
MENU FILENAME SCR_350x60a0x196
MENU OPEN-FILE MENU CHANNEL IS-WORK RED PAPER CLS
3 WHITE BORDER MENU CHANNEL IS-OUTPUT

FCB DATOS
DATOS FILENAME SCR_162x60a350x196
DATOS OPEN-FILE DATOS IS-WORK WHITE PAPER CLS
3 RED BORDER DATOS CHANNEL IS-INPUT
```

Con estas líneas la pantalla queda dividida en 3 ventanas unidas a 3 canales, cada uno de ellos con un nombre por el que se le puede referenciar: DIBUJO, la ventana mayor en la que se realizan los dibujos; MENU, una ventana más pequeña en la zona inferior izquierda para la presentación de menús y DATOS, que es la ventana menor en la zona inferior derecha para la entrada de datos.
En una próxima ocasión os enviaré algunas palabras FORTH pertenecientes al mismo programa y que se pueden considerar como extensiones del lenguaje para gráficos en tres dimensiones.

José Carlos de Prada
MADRID QLave (216)

SOBRE LEER DISCOS EN FORMATO MAC EN EL QL.

Lo primero de todo decir que es imposible, pero voy a contar toda la historia de

un experimento de leer/escribir en formato Mac en un QL. Para hacerlo pedí información y que me grabaran un disco en formato MAC a Serafín Olcoz y José Manuel Garrido Dobón (Ambos usuarios de MAC, pero viejos usuarios QL). Los resultados son :

- Con una utilidad llamada CPMCOPY (escrita en Superbasic y compilada con el Qliberator), al no enviar suficiente información Serafín tuve que usar la opción analizar el disco para conocer la mayor parte del formato o todo el formato (es una utilidad bastante inteligente y se equivoca muy poco). La respuesta de analizar el disco fue bastante rápida, solamente unos 3 segundos, y es :

Unable to read Track 0
It could be unformatted support access to disks of this density and/or sector size or the disk interface does not have direct sector access facilties.

- Con una utilidad escrita en CP/M, Analyse (proviene del emulador CP/M SUCCESS, uno de los cuatro que posee el QL). El resultado me llegó más rápido todavía, solamente 2 segundos escasos :

Sorry, The disk controller cannot read this disk.

Como podran observar, el experimento ha sido un auténtico fracaso a medias, pues he podido demostrar que el QL no puede leer el formato Mac.

Hasta ahora solamente sabia que mi interface de disco no podia controlar las nuevas unidades de disco 3.5" HD, que permiten un almacenamiento de datos de 1.2Mb. Pero por lo visto Apple usa un interface especial de disco (que controla la velocidad del motor), y una unidad de disco 3.5" no Standard (El Mac cuando no necesita un disquette lo expulsa y ya está. He buscado una antigua foto del Mac y no he encontrado un botón para sacar el disco manualmente. ¿ Existe o no existe ? esa es la cuestión.).

También he preguntado a Serafín si el Mac podria leer los discos QL, pero me ha dicho que ese software no se ha hecho aún para el Mac (sé que se ha hecho para leer discos IBM en el Mac y viceversa, pero se necesita software y Hardware, y lo peor de todo, es bastante carillo). El QL podria leer los discos Mac si se le agrega el interface controlador/disco Apple y el software necesario para que eso funcione, pero creo que es más barato e interesante comprar el Mac versión base y comunicarlos via RS232.

Para pasar ficheros QUILL a Mac via RS232, lo primero que hay que hacer es instalar una impresora Standard (no preambulo, postabumbulo, etc....), caracteres normales (no negrita, subrayado,etc..), no pie de página,.... Y usar la opción imprimir para crear un fichero _LIS. Si usamos un editor, el fichero final seria ASCII puros y no habria ningún problema.

Como todos sabeis, soy un supercoleccionista y tengo almacenada mucha información. He buscado, y he encontrado en la página 144 de la revista Byte de Mayo 88 el paquete DAYNA FILE (Dayna Comunication Inc), que consiste en leer discos en formato IBM PC/PS en un Mac. El Mac debe tener un port SCSI. La configuración drive 5.25" 360K cuesta 595 dólares, y la drive 5.25" 360K y drive 3.5" 720K cuesta 849 dólares. Algo carillo comparado con mi programa IBMCOPY que cuesta 20 libras con el CPMCOPY y posee la característica de traducir los caracteres ASCII no Standard usados para caracteres extrangeros. Y lo mejor de todo, un Mac con DAYNA FILE es una presa fácil para un francotirador QL. Psst... la época imperial ya es historia, estamos en retirada estratégica (un usuario de QL tiene más moral que el alcoyano).

Como habran observado mis inteligentes lectores, el MAC necesita para poder escribir en un formato Alien una unidad de discos Standard ( su unida de discos lo único que tiene en común con una Standard es que usan los mismos discos 3.5" DS/DD del mercado. Asi de fácil).

Salvador Merino
Fuengirola (QLave-154).